

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



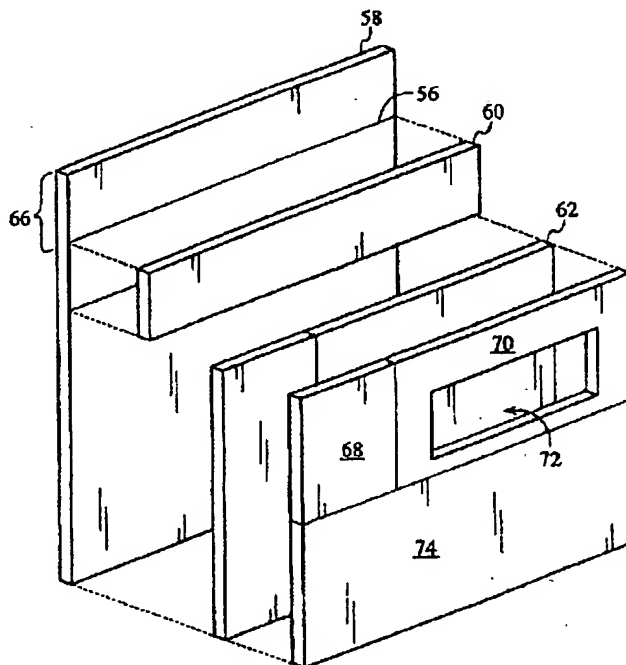
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 5/01, 15/00, 17/30, 3/14		A1	(11) International Publication Number: WO 99/40507
			(43) International Publication Date: 12 August 1999 (12.08.99)
(21) International Application Number: PCT/US99/02512		(74) Agents: BROOKS, Kenneth, C. et al.; Townsend and Townsend and Crew LLP, 8th floor, Two Embarcadero Center, San Francisco, CA 94111-3834 (US).	
(22) International Filing Date: 5 February 1999 (05.02.99)			
(30) Priority Data: 09/020,098 6 February 1998 (06.02.98) US		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application US 09/020,098 (CON) Filed on 6 February 1998 (06.02.98)			
(71) Applicant (for all designated States except US): MANNING & NAPIER INFORMATION SERVICE [US/US]; 1100 Chase Square, Rochester, NY 14604 (US).			
(72) Inventors; and (75) Inventors/Applicants (for US only): ANDREWS, Taggard, W. [US/US]; 1100 Chase Square, Rochester, NY 14604 (US). CHRONIS, Todd [US/US]; 63 Crossman Terrace, Rochester, NY 14620 (US). GOTO, Makoto [US/US]; 1100 Chase Square, Rochester, NY 14604 (US). POGODA-CURTIS, Stacey [US/US]; 1100 Chase Square, Rochester, NY 14604 (US).		Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.	

(54) Title: METHOD OF UPDATING DISPLAY FRAMES WHILE PRESERVING INFORMATION ASSOCIATED THEREWITH

(57) Abstract

A method and apparatus for resizing or re-drawing windows (58) without requiring additional server calls to supply the necessary information objects to dynamically update the content associated therewith. The invention does so by dividing a window into a plurality of frames (60 and 62), defining a reference window (56), and initializing the information objects when the reference window (56) is loaded into a client's computer (12). In this fashion, the information objects are associated with a subset of the plurality of memory address locations of the RAM (28) in the client's computer (12) with which the reference window (56) is associated. This prevents overwriting the memory address locations in which the information objects are located, because the lifetime of a reference window is as long as the same is open. Thus, information objects present in the memory address locations associated therewith are not considered freed-up by the browser (54) until the reference window (56) is closed.



BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MIW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

METHOD OF UPDATING DISPLAY FRAMES WHILE PRESERVING INFORMATION ASSOCIATED THEREWITH

COPYRIGHT NOTICE

A portion of the disclosure recited in the specification contains material which is subject to copyright protection. Specifically, a source code appendix is included that lists instructions for a process by which the present invention is practiced in a computer system. The copyright owner has no objection to the facsimile reproduction of the specification as filed in the Patent and Trademark Office. Otherwise all copyright rights are reserved.

BACKGROUND OF THE INVENTION

The present invention relates to programmable computer systems. More particularly, the present invention is directed to a computer system ideally suited for efficient data communication over a data network.

Data networks, such as the "Internet," typically include a plurality of users using client computers communicating with a remote server computers to transfer information therebetween. To facilitate the transfer, the users employ a computer system having a "web" browser that provides graphical user interface (GUI)-based communication with a "web page" obtained from a server. One popular collection of servers uses a standardized Hypertext Transfer Protocol (HTTP) to provide information and is known as the "World Wide Web." The information is presented as web pages written as text with standardized formatting and control symbols known as HyperText Mark-up Language (HTML). HTML provides basic document formatting and allows a server to specify "links" to other servers and files. Use of an HTML-compliant browser involves specification of a link via a Uniform Resource Locator (URL). Upon such specification, the user's client computer makes a TCP/IP request to the server identified in the link and receives an HTML file that is

interpreted by the browser so that an electronic HTML document made up of one or more web pages may be displayed on the client's computer.

The HTML markup language is analogous in some ways to the formatting codes used in word processing documents. A word processing document viewed through a word processing program is actually a combination of a visual representation of text and a series of non-visually perceivable formatting codes (e.g., carriage return, bold, underline) which instruct the word processing program to display the word processing document in a specified way. Similarly, an HTML file is actually a combination of the visually perceivable text, referred to as an HTML document and a series of non-visually perceivable "tags" or "anchors" (for new paragraphs, graphics images, hypertext links, etc.) which instruct the browser program to display the hypertext document in a specified way.

A HTML file is usually broken down into sections, with each section delineated by one or more HTML tags. HTML tags are formatting codes surrounded by the characters < and > (less than and greater than symbols). Some HTML tags have a start tag and an end tag. In general, end tags are in the format </"symbol"> where the "symbol" is the character string found between the characters < and > in the start tag. FIG. 1 is an example of a series of HTML document tags forming a template for a typical hypertext document using the tags <html> and </html>. Then the "head" of the document, which typically includes a title, is defined using the tags <head>, </head>, <title>, and </title>, respectively. Following the head comes the "body" of the document which is often organized into subtopics with different levels of headings. The body is defined by the tags <body> and </body>. Headings are indicated by the tags <h#> and </h#>, where # is the level of the heading. Heading levels indicate the relative size of the heading. Heading level 1 is the largest heading size and heading level 6 is the smallest heading size. Finally, it is good practice to indicate the author of the document at the bottom of the document using the tags <address> and </address>.

Once the HTML template has been established, text is added to create a basic HTML document. In order to improve readability, character and paragraph formatting tags may be added to the HTML file. For example, the <p> tag

instructs the browser to begin a new paragraph. If an author wants to highlight some text in bold, the author inserts the `` tag at the beginning of the text to be highlighted and inserts a `` tag at the end of the text to be highlighted. The tags `<i>` and `</i>` indicate text to display in italics.

5 To provide additional versatility to HTML files, additional tags have been provided to allow "linking" to additional files. If a reader of a HTML document wants to know more about a topic before reading the rest of the current HTML document, the reader selects a "link" or "hot link", which retrieves a new HTML file to display a new HTML. The link could take the reader to a second HTML file (i.e.,
10 a "destination document") having the same URL.

 A hot link to a destination document is made by placing a "reference anchor" around the text to be highlighted and then providing a network location, typically using a URL, where the destination file is located. A service type is a required part of a URL. The service type tells the client's browser how to contact the
15 server for the requested data. The most common service type employed is the Hypertext Transport Protocol or http. The web can handle several other services including gopher, wais, ftp, netnews, and telnet and can be extended to handle new service types. A system name is also a required part of a URL. The system name is the fully qualified domain name of the server which stores the data being requested.
20 A port is an optional part of a URL. Ports are the network socket addresses for specific protocols. By default, http connects at port. Ports are only needed when the server does not communicate on the default port for that service. A directory path is a required part of a URL. Once connected to the system in question, a path to the file must be specified. A filename is an optional part of a URL. The file name is the
25 data file itself. The server can be configured so that if a filename isn't specified, a default file or directory listing is returned. A search component is another optional part of a URL. If the URL is a request to search a data base, the query can be embedded in the URL.

 HTML files have been traditionally used with common Information
30 Processing Common Gateway Interface (CGI) script to pass client requests to an information engine on a server. The server typically includes an information engine

that retrieves multiple files or real-time news. Specific implementations of the information engine include, without limitation, conventional boolean search engines such as those produced by Digital Equipment Corporation, information engines employing natural language processing such as DR-LINK produced by Manning & Napier Information Services, news retrieval systems such as those produced by Pointcast Technologies, Inc., and the like.

A drawback with CGI script is that dynamic creation of HTML documents by the client's computer is not facilitated. This proves problematic with interactive HTML documents such as a HTML document that includes computational functions associated with calculator programs. For each computation performed on the web page, the server must be contacted to perform the same. Each time a the server is contacted, a varying delay occurs before the resulting HTML document showing the computational results is returned to the client's computer. This delay is typically on the order of seconds and represents the major delay, and, arguably, the major drawback of efficiently obtaining information from today's WWW.

A relatively new scripting language, JavaScript overcomes many of the drawbacks associated with CGI script. Specifically, JavaScript allows executable content to be embedded in HTML files. In this fashion, JavaScript allows programmatic control over the browser, as well as the content of HTML documents. This facilitates dynamically updating the content of HTML documents on the client computer, thereby significantly reducing the wait-time when invoking a server call. For purposes of the present invention, dynamically created content is information which is derived from code being executed by the browser. Typically, a visual representation of an HTML document is in a browser window having a plurality of pixels associated therewith which correspond to a subset of the memory address location of the client computer's RAM. Information objects and the properties thereof, such as variables, necessary for dynamically updating the content of an HTML document are typically present in RAM so they can be accessed quickly as described in chapters 11 and 12 of JavaScript, The Definitive Guide, published by O'Reilly and Associated (2nd ed.). A problem exists with JavaScript, however, in that access to the information objects, as well as the properties thereof, is often lost

when the browser window is redrawn or refreshed, such as when a window or frame displaying a HTML page that uses the information objects is resized or redisplayed. In order for the client's computer to reacquire the information objects after a resizing or redrawing operation, an additional server call is required to reload information objects that are necessary to display the pages within the window and/or frames, thereby slowing the overall communication between the time that the client requests the information objects and the time in which the same are received.

One prior art method to retain information objects is to use an "invisible frame." The invisible frame is not displayed on a user's screen but, instead, is used to hold objects that other visible frames need to display their pages. Such a method is described at pp. 204-205 of the O'Reilly reference, *supra*, but often necessitates additional server calls to update frames when the same is resized or redrawn.

What is needed, therefore, is an apparatus and method for resizing or redrawing windows without requiring additional server calls to supply the necessary information objects to dynamically update the content associated therewith.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for resizing or redrawing windows without requiring additional server calls to supply the necessary information objects to dynamically update the content associated therewith. The invention does so by dividing a window into a plurality of frames, defining a reference window, and initializing all the information objects when the reference window is loaded into the client's computer. The invention is based upon the discovery that by employing the invisible frame technique, user defined variables associated with the information objects that have been dynamically updated are reset to initial values if the window containing the information objects is redrawn and/or resized. In the present invention, the information objects and the variables associated therewith, are associated with a subset of the plurality of memory address locations of the RAM with which the reference window is associated. This prevents overwriting the memory address locations in which the information objects and the properties

associated therewith, such as variables, are located, because the lifetime of a reference window is as long as the same is open. Thus, information objects present in the memory address associated therewith are not considered freed-up by the browser until the reference window is closed. All information associated with the frames are loaded outside of the memory address locations associated with the reference window. In this fashion, content of the frames may be dynamically updated by populating the frames with the information objects associated with the reference window. Thus, when a frame is resized or moved, such as when the reference window is resized or moved, the information objects necessary to redraw the frames are maintained in the client's computer, as well as the properties associated with these information objects, such as user defined variables.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an example of a series of HTML file tags forming a template for a HTML file of the prior art.

Fig. 2 is simplified plan view of a computer network in which the present invention is implemented.

Fig. 3 is a block diagram of a client computer shown above in Fig. 2.

Fig. 4 is plan view of a reference window having a visual representation of an HTML document shown therein in accord with the present invention.

Fig. 5 is an exploded perspective view of the reference window, shown above in Fig. 4 in relation to a top level window of a client-side browser.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Fig. 2, a large area network is shown, such as the Internet, which includes a plurality of networked "servers" 10 that are accessible by "clients" 12. Communication between the servers 10 and the clients 12 typically occurs over a publicly accessible network, such as a public switched telephone network over ASDL telephone lines or large bandwidth trunks, such as T1 or OC3 service. The clients 12 access the various servers 10 through an Internet service

provider, e.g., America On-Line, Prodigy, CompuServe and the like, by executing application specific software, commonly referred to as a "browser", on a computer 14, shown more clearly in Fig. 3.

Referring to Fig. 3, the computer 14 includes a system unit 20 having one or more system buses 22 placing various components of the system in data communication. For example, a microprocessor 24 is placed in data communication with both a read only memory (ROM) 26 and random access memory (RAM) 28 via the system bus 22. The ROM 26 contains among other code, the Basic Input-Output system (BIOS) which controls basic hardware operation such as the interaction with peripheral components such as disk drives 30 and 32, as well as the keyboard 34. The RAM 28 is the main memory into which the operating system and application programs are loaded. The memory management chip 36 is in data communication with the system bus 22 to control direct memory access (DMA) operations. DMA operations include passing data between the RAM 28 and the hard disk drive 30 and the floppy disk drive 32. Also in data communication with the system bus 22 are various I/O controllers: a keyboard controller 38, a mouse controller 40, a video controller 42, and an audio controller 44. The keyboard controller 38 provides a hardware interface for the keyboard 36, the mouse controller 40 provides the hardware interface for a mouse 46, or other point and click device, and the video controller 40 provides a hardware interface for a display 48. A modem 50 enables data communication over the network. The operating system 52 of the computer 14 may be DOS, WINDOWS 3.x, WINDOWS 95, OS/2, or other known operating system. The RAM 28 also supports a number of Internet access tools, including, for example, an HTTP-compliant web browser having a JavaScript interpreter, such as Netscape Navigator 3.0, Microsoft Explorer 3.0 and the like.

Referring to both Figs. 2 and 3, to access information over the Internet, the browser 54 employs a tcp/ip connection to pass a request to a server 10 running an HTTP "service" (under the WINDOWS operating system) or a "daemon" under the UNIX operating system. The request is typically achieved employing a protocol that can be used to communicate between the server 10 and the client 12. The HTTP server 10 then responds to the protocol, typically by sending a "web

page" formatted as an HTML file. The browser 54 interprets the HTML file and may form a visual representation of the same using local resources, e.g., fonts and colors. In the preferred embodiments, the aforementioned protocol comprises of scripting language such as JavaScript.

5 Referring to Figs. 3 and 4, upon interpreting an HTML file, the browser 54 provides a visual representation of an HTML document 56, among a plurality of rectangular pixel regions on the display of the client's computer 14, defining a browser region shown as 58 in Fig. 5. The rectangular pixel regions can be segmented into windows or frames, each displaying an associated HTML page or other information. The source code corresponding to the HTML document 56 is shown in the Appendix which is attached hereto and has embedded therein, script language that incorporates a plurality of information objects. For purpose of the present invention, information objects may have associated therewith, functions, variables, arrays and other objects, data or executable code. Examples of information
10 objects are as follows:

```
20 / User Information
var UserID = "drlddev";
var PID = "4728";
var currentInfoView = 0;
var currentProjectID = "";
var currentRequestName = "";
var currentDateTimeStamp = "";
var currentMlmDateTimeStamp = "";
25 var PRPRUser = false;

// User Defaults
var def = new Object();
30 def.Email = "mak@textwise.com";
def.query = "I would like information about ";
def.dateRange = "all";
def.fromDate = "01/01/60";
def.toDate = "11/25/97";
```

35 One of the objects in the script language is a Window object which is the first object created by the browser when a web page is located. The Window object supports,

inter alia, methods to pop-up dialog boxes to display messages to obtain input from the client 12, as well as provides a method by which to create new browser windows 58 and frames, such as 62, without necessitating an additional server call. To that end, the Window object includes methods, such as alert and prompt, which allow
5 programs for the window to load a new document, including documents present at a differing URL. In this fashion, dynamic generation of an HTML document may be achieved.

Referring to Figs. 3, 4 and 5, the approach of the present invention to preserve information objects and the properties associated therewith, such as
10 variables, even upon resizing or redisplaying a window or frame is to create a plurality of frames 60 and 62 associated within the window 56, with the window 56 defining a reference window. The reference window 56 is typically displayed within a top level window 58 of the browser 54 in which the uppermost portion includes a menu bar 66. Each of the frames 60 and 62 consists of a subset of the plurality of
15 pixels associated with the reference window 56, mutually exclusive of the subset of the plurality of pixels of the remaining frames. In this fashion, each of the frames 60 and 62 is associated with a mutually exclusive subset of the plurality of address locations in the RAM 28. In the exemplary embodiment of the present invention, a plurality of subframes 68, 70, 72 and 74 are associated with frame 62.

20 In the present invention, for example, frame 60 is a static frame in that no information is updated without necessitating an additional server call. To update the content in any of the remaining frames, such as frame 62 comprising subframes 68, 70, 72 or 74, the code necessary to update these frames would be associated with frame 60. In this fashion, the content of frame 62, or any of its corresponding
25 subframes 68, 70, 72 or 74, may be updated without overwriting the code necessary to carry out the update method.

A problem was encountered upon resizing or moving the reference window 56, i.e., having the reference window 56 corresponding to a different subset of the plurality of pixels on the display 48. This resulted in the information objects,
30 and the properties associated therewith such as user defined variables, associated with the frames 60 and 62 becoming lost. Specifically, it was recognized that the

information objects were lost in a manner similar to when frames have content updated, i.e., the memory address locations of the RAM 28 storing the information objects were overwritten. This problem would occur when the window was resized, moved, or associated with a new URL.

5 To avoid this problem, information objects are stored in the memory address locations of the RAM 28 associated with information of the HTML file that creates the reference window 56. This is accomplished by defining the information objects in the window 56 definition using the HTML of Appendix A. The information objects are initialized in the reference window 56 in which the frames are
10 defined upon the HTML file being written to RAM 28. This procedure stores the information objects at memory addresses which the browser 54 does not allow to be overwritten until the reference window 56 is closed. This minimizes the number of server calls necessitated when updating the content of a frame, because the information objects necessary to achieve the same are resident in the client's
15 computer 14 at RAM 28. The information objects are populated in the frames 60, 62, 68, 70, 72 or 74, when an updating is required.

Thus, the invention has been described in connection with specific embodiments thereof. However, nothing in the specification shall be construed to limit the scope of the invention. Rather, the invention is determined solely by the appended claims.

APPENDIX

```
<HTML>
<HEAD>
<TITLE>DR-LINK</TITLE>
<BASE Href=http://host149.mnis.net:8082/DRLv3stat/>
<SCRIPT Language='javascript'>

//-----
opener.location = 'http://www.mnis.net';

//---Variables & Arrays-----

var HLwindow = false;
var HelpWindow = false;

var savedRequest = false;
var bgColor = "ccccff";
var textColor = "000066";

var visNames = new Array("Bar Chart Visualizer");
var visShortNames = new Array("BarChart");

var today = new Date();~
var justDate = (today.getMonth() +1) + "/" + today.getDate() + "/"
+ today.getYear();

// User Information

var UserID = "drldev";
var PID = "20442";
var currentInfoView = 0;
var currentProjectID = "";
var currentRequestName = "";
var currentDateTimeStamp = "";
var currentMlmDateTimeStamp = "";

// User Defaults

var def = new Object();
def.Email = "mak@textwise.com";
def.query = "I would like information about ";
def.dateRange = "all";
def.fromDate = "01/01/60";
def.toDate = "11/25/97";
def.DBs = "";
def.docsRet = 100;
def.view = "Rel";
def.sort = "Ran";
def.alertSend = "hl";
def.alertPrecision = "high";
def.alertFrequency = "twice";
def.alertDocsRet = "5";
```

```
def.printIncQuery = true;
def.printIncDbs = true;
def.printIncDates = true;
def.printContent = "hl";
def.printDocs = "marked";
def.Highlight = "AllHL";
def.visDoc = "marked";
def.visType = "BarChart";

var currentQuery = def.query;
var UserEmail = def.Email;
var currentDateRange = def.dateRange;
var currentFromDate = def.fromDate;
var currentToDate = def.toDate;
var currentDocsRequested = def.docsRet;
var currentResView = def.view;
var currentResSort = def.sort;
var currentAlertSend = def.alertSend;
var currentAlertPrecision = def.alertPrecision;
var currentAlertFrequency = def.alertFrequency;
var currentAlertDocsRet = def.alertDocsRet;
var currentPrintIncQuery = def.printIncQuery;
var currentPrintIncDbs = def.printIncDbs;
var currentPrintIncDates = def.printIncDates;
var currentPrintContent = def.printContent;
var currentPrintDocs = def.printDocs;
var currentHighlight = def.Highlight;
var currentVisDocs = def.visDoc;
var currentVisType = def.visType;

// Query Review

var QRURL;
var AnalInfo;
var CEDim;
var AttribQ;
var Pred;
var EvalCom;
var TFPast;
var TFPresent;
var TFFuture;
var TFAI;
var UserManTerms;
var ManTerms;
var ProperNouns;
var TermAlternatives;
var Subjects;
var HiddenPN;

// Results

var haveResults = false;
```

```
var QueryKey;
Result = new Array();
ResSubject = new Array();
ResDatabase = new Array();

var ResQuery;
var ResFromDate;
var ResToDate;
var ResDBs;
var ResMlm = false;

Mlm = false;

var CurrentResult = 0;
var CurrentRank = 0;
var resultIndex = new Array();

// More Like Marked

OldResult = new Array();
OldResSubject = new Array();
OldResDatabase = new Array();

// Alerts

var haveAlertsResults = false;

AlertDates = new Array
('11/10/97', '11/11/97', '11/12/97', '11/13/97', '11/14/97', '11/15/97',
'11/16/97', '11/17/97', '11/18/97', '11/19/97', '11/20/97', '11/21/97',
'11/22/97', '11/23/97', '11/24/97', '11/25/97', '11/26/97', '11/27/97',
'11/28/97', '11/29/97', '11/30/97', '12/01/97', '12/02/97', '12/03/97',
'12/04/97', '12/05/97', '12/06/97', '12/07/97', '12/08/97', '12/09/97');
var beginAlertDate = AlertDates.length - 1;
var endAlertDate = AlertDates.length - 1;

Alerts = new Array();
var currentAlert = -1;

AlertsRes = new Array();
var currentAlertResIndex = 0;

var alertQuery;
var alertDBs;

var alertsIndexArray;

var alertsDateTimeStamp;

// Help
```

```
var currentHelpIndex = 0;

// Databases

Database = new Array();
Selected = new Array();

Database[0] = new Object();
Database[0].name = "Aerospace/Defense IntellX";
Database[0].value = 80;
Database[0].checked = false;
Database[0].money = true;

Database[1] = new Object();
Database[1].name = "AFX-ASIA";
Database[1].value = 472;
Database[1].checked = false;
Database[1].money = false;

Database[2] = new Object();
Database[2].name = "AFX-EUROPE";
Database[2].value = 6;
Database[2].checked = false;
Database[2].money = false;

Database[3] = new Object();
Database[3].name = "Agriculture - MNIS";
Database[3].value = 809;
Database[3].checked = false;
Database[3].money = false;

Database[4] = new Object();
Database[4].name = "Auto IntellX";
Database[4].value = 75;
Database[4].checked = false;
Database[4].money = true;

Database[5] = new Object();
Database[5].name = "Banking - MNIS";
Database[5].value = 798;
Database[5].checked = false;
Database[5].money = false;

Database[6] = new Object();
Database[6].name = "Banking IntellX";
Database[6].value = 76;
Database[6].checked = false;
Database[6].money = true;

Database[7] = new Object();
Database[7].name = "Biotech IntellX";
Database[7].value = 77;
Database[7].checked = false;
```



```
Database[7].money = true;

Database[8] = new Object();
Database[8].name = "Computer IntellX";
Database[8].value = 78;
Database[8].checked = false;
Database[8].money = true;

Database[9] = new Object();
Database[9].name = "Computer Sciences & Engineering - CSA";
Database[9].value = 836;
Database[9].checked = false;
Database[9].money = false;

Database[10] = new Object();
Database[10].name = "Computers, Software & Technology - MNIS";
Database[10].value = 157;
Database[10].checked = false;
Database[10].money = false;

Database[11] = new Object();
Database[11].name = "Construction IntellX";
Database[11].value = 79;
Database[11].checked = false;
Database[11].money = true;

Database[12] = new Object();
Database[12].name = "Consumer News - Comline";
Database[12].value = 794;
Database[12].checked = false;
Database[12].money = true;

Database[13] = new Object();
Database[13].name = "Electronics - Comline";
Database[13].value = 793;
Database[13].checked = false;
Database[13].money = true;

Database[14] = new Object();
Database[14].name = "Energy IntellX";
Database[14].value = 81;
Database[14].checked = false;
Database[14].money = true;

Database[15] = new Object();
Database[15].name = "Entertainment IntellX";
Database[15].value = 82;
Database[15].checked = false;
Database[15].money = true;

Database[16] = new Object();
Database[16].name = "Environment/Waste IntellX";
Database[16].value = 83;
Database[16].checked = false;
```

```
Database[16].money = true;

Database[17] = new Object();
Database[17].name = "Food & Beverage IntellX";
Database[17].value = 84;
Database[17].checked = false;
Database[17].money = true;

Database[18] = new Object();
Database[18].name = "General Business IntellX";
Database[18].value = 85;
Database[18].checked = false;
Database[18].money = true;

Database[19] = new Object();
Database[19].name = "Government IntellX";
Database[19].value = 100;
Database[19].checked = false;
Database[19].money = true;

Database[20] = new Object();
Database[20].name = "Healthcare IntellX";
Database[20].value = 86;
Database[20].checked = false;
Database[20].money = true;

Database[21] = new Object();
Database[21].name = "IAC Market Collection (1997)";
Database[21].value = 808;
Database[21].checked = false;
Database[21].money = true;

Database[22] = new Object();
Database[22].name = "IAC Market Research (1996)";
Database[22].value = 806;
Database[22].checked = false;
Database[22].money = true;

Database[23] = new Object();
Database[23].name = "IAC Newsletter Collection (1995)";
Database[23].value = 857;
Database[23].checked = false;
Database[23].money = true;

Database[24] = new Object();
Database[24].name = "IAC Newsletter Collection (1996)";
Database[24].value = 155;
Database[24].checked = false;
Database[24].money = true;

Database[25] = new Object();
Database[25].name = "IAC Newsletter Collection (1997)";
Database[25].value = 807;
Database[25].checked = false;
```

```
<td align=right valign=top> <a href=javascript:enterQuery();>  </a> </td>
```

```
</tr>
```

```
</table>
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

WHAT IS CLAIMED IS:

- 1 1. A method for transferring data in a script-based
2 communications system having a display, said method comprising the steps of:
3 receiving information including an HTML file and executable code
4 having a plurality of information objects contained therein;
5 forming a visual representation of said HTML file on said display,
6 defining a reference window, said window having a plurality of pixels;
7 creating, from a subset of said plurality of pixels, an HTML document,
8 defining a frame; and
9 dynamically updating said HTML document by invoking one of said
10 plurality of information objects while preserving a subgroup of said plurality of
11 information objects.
- 1 2. The method as recited in claim 1 wherein said system includes a
2 memory having a plurality of address locations and said updating step includes storing
3 said subgroup of information in a subset of said plurality of address locations
4 associated with said reference window.
- 1 3. The method as recited in claim 1 wherein said updating step
2 includes a step of initializing said plurality of information objects in said reference
3 window.
- 1 4. The method as recited in claim 1 wherein said updating step
2 includes associating said frame with a different URL.
- 1 5. The method as recited in claim 1 wherein said updating step
2 includes associating said frame with a differing subset of pixels.
- 1 6. The method as recited in claim 1 wherein said updating step
2 includes resizing said frame.

1 7. The method as recited in claim 1 wherein said creating step
2 creates a plurality of HTML documents, each of which is associated with a group of
3 pixels of said plurality of pixels, mutually exclusive of pixels associated with the
4 remaining HTML documents.

1 8. The method as recited in claim 1 wherein said script-based data
2 includes JavaScript.

1 9. The method as recited in claim 2 wherein said subgroup
2 includes all of said plurality of information objects.

1 10. A method to facilitate data communication within a script-based
2 system including a visual display and a memory having a plurality of address
3 locations, said method comprising:
4 receiving an HTML file having executable code embedded therein,
5 said executable code including a plurality of information objects;
6 creating a visual representation of said text information, defining a
7 reference window;
8 storing a subgroup of said plurality of information objects among a
9 subset of said plurality of address locations, with said subset being associated with
10 said reference window;
11 forming, in said reference window, a frame having content associated
12 therewith;
13 dynamically updating said content by referencing at least one
14 information object associated with said reference window.

1 11. The method as recited in claim 10 wherein said updating step
2 includes a step of initializing said plurality of information objects in said reference
3 window.

1 12. The method as recited in claim 10 wherein said updating step
2 includes associating said frame with a different URL.

1 13. The method as recited in claim 10 wherein said updating step
2 includes associating said frame with a differing subset of pixels.

1 14. The method as recited in claim 10 wherein said updating step
2 includes resizing said frame.

1 15. The method as recited in claim 10 wherein said creating step
2 creates a plurality of HTML documents, each of which is associated with a group of
3 pixels of said plurality of pixels, mutually exclusive of pixels associated with the
4 remaining HTML documents.

1 16. The method as recited in claim 10 wherein said script-based
2 data includes JavaScript.

1 17. The method as recited in claim 10 wherein said subgroup of
2 said plurality of information objects includes all of said plurality of information
3 objects.

1 18. In a processing system of the type including a processor, a
2 memory in data communication with said process and a display having a plurality
3 pixels, with said memory having a plurality of address locations, a subset of which
4 stores an HTML file and executable code having a plurality of information objects
5 contained therein, with said source code having a first subroutine adapted to form a
6 visual representation of said HTML file on said display, defining a reference window,
7 said window having a plurality of pixels, a second subroutine adapted to create from a
8 subset of said plurality of pixels, an HTML document, defining a frame, wherein an
9 improvement comprises of a third subroutine stored in said memory which is adapted
10 to dynamically update said HTML document by invoking one of said plurality of

11 information objects while preserving a subgroup of said plurality of information
12 objects.

1 19. The system as recited in claim 18 wherein said third subroutine
2 is further adapted to store subgroup of information objects in a subset of said plurality
3 of address locations associated with said reference window.

1 20. The system as recited in claim 18 wherein said third subroutine
2 is further adapted to store all of said information objects in a subset of said plurality
3 of address locations associated with said reference window.

```
<html>
<head>
<title>

</title>
</head>
<!--this is a comment-->
<body>

<address>

</address>
</body>
</html>
```

FIG. 1

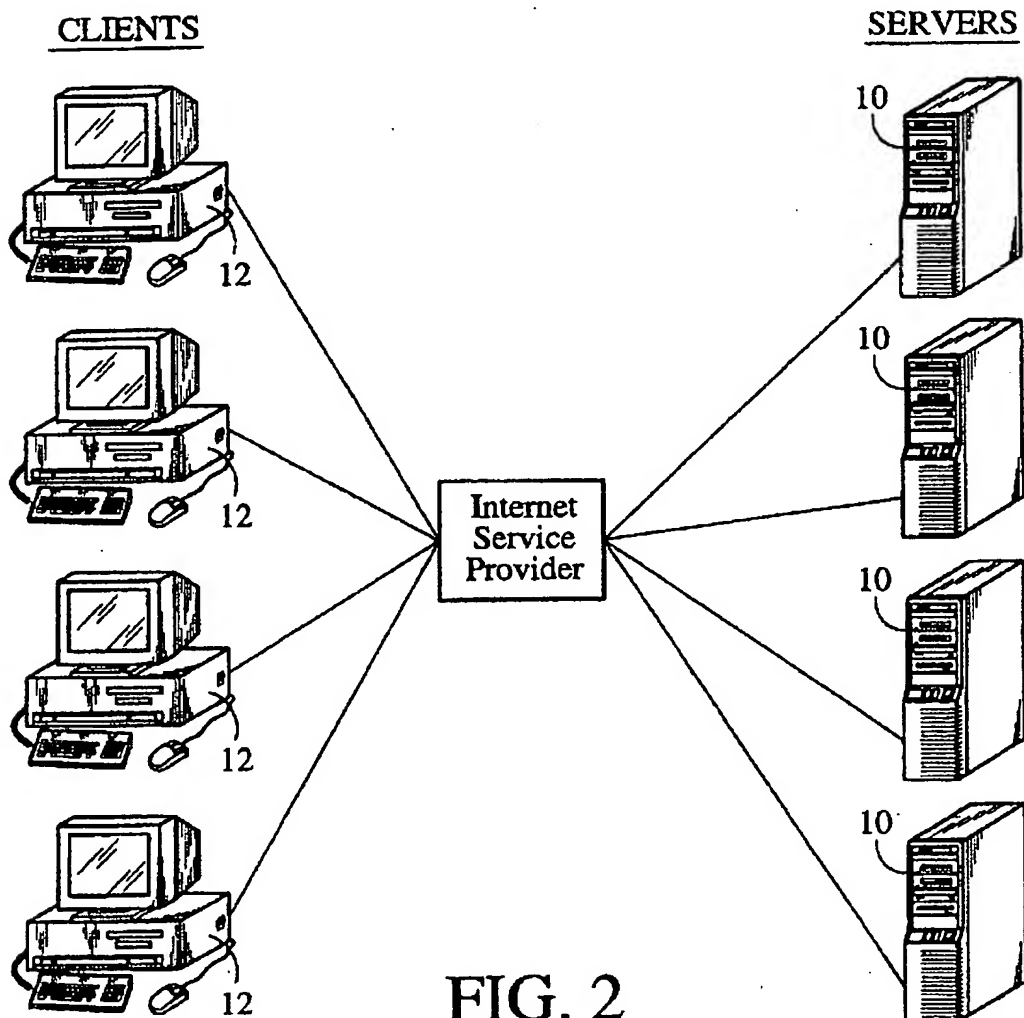


FIG. 2

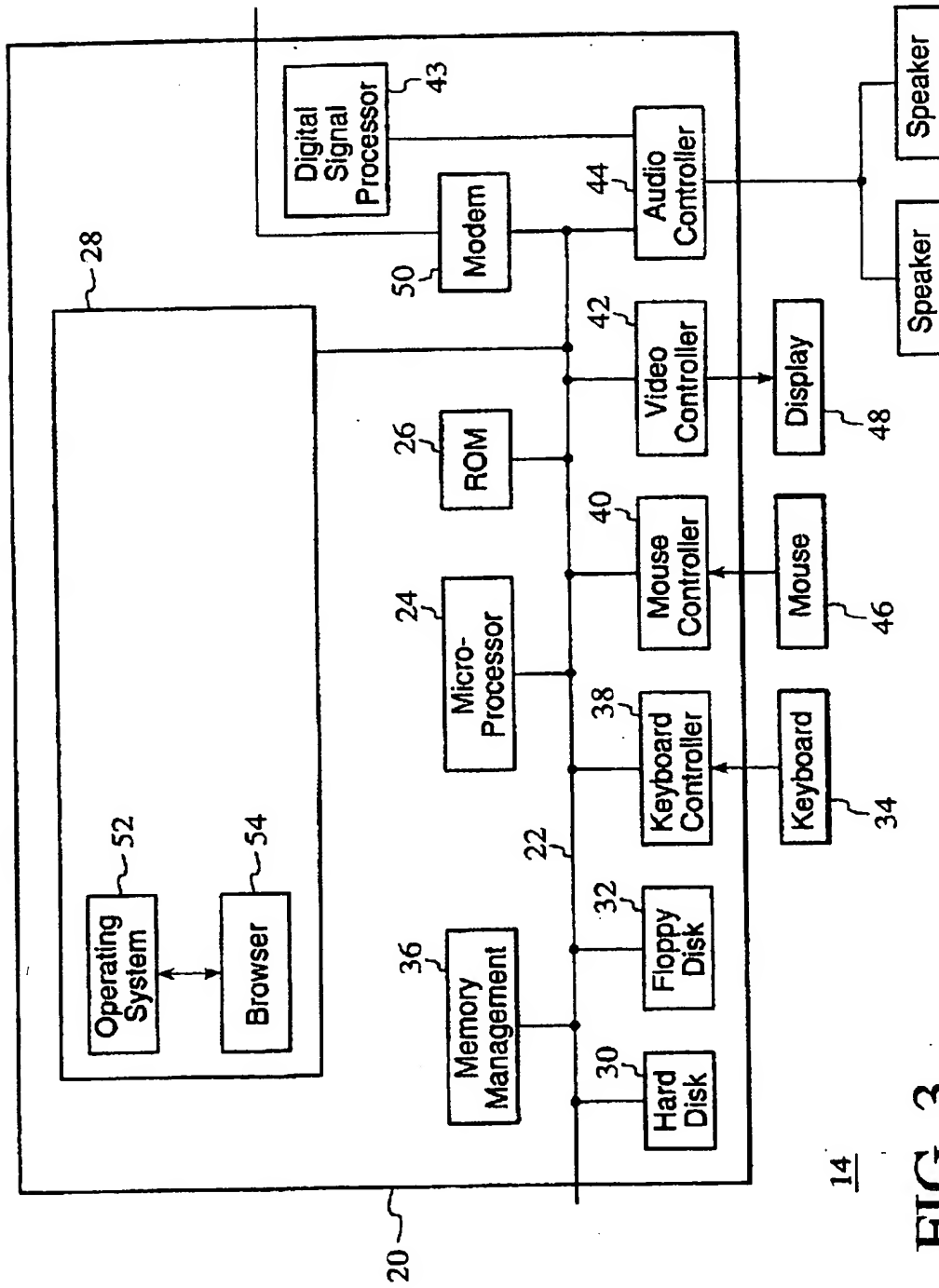


FIG. 3

DR-Link		Request Agent	Saved Requests	Dr-Link Alerts	User Prefs	Contact MNIS	56	Stop
Request: I would like information about 68		Select Databases						
<input checked="" type="radio"/> Request <input type="radio"/> Dates <input type="radio"/> Databases		Select Dates						
View Request History New Request		Type your request into the box below. Please capitalize the first letter of all proper nouns (e.g., "India"). I would like information about 72						
		Project ID:		Documents Requested:		100		
		74						
Help		Start Search						

FIG. 4

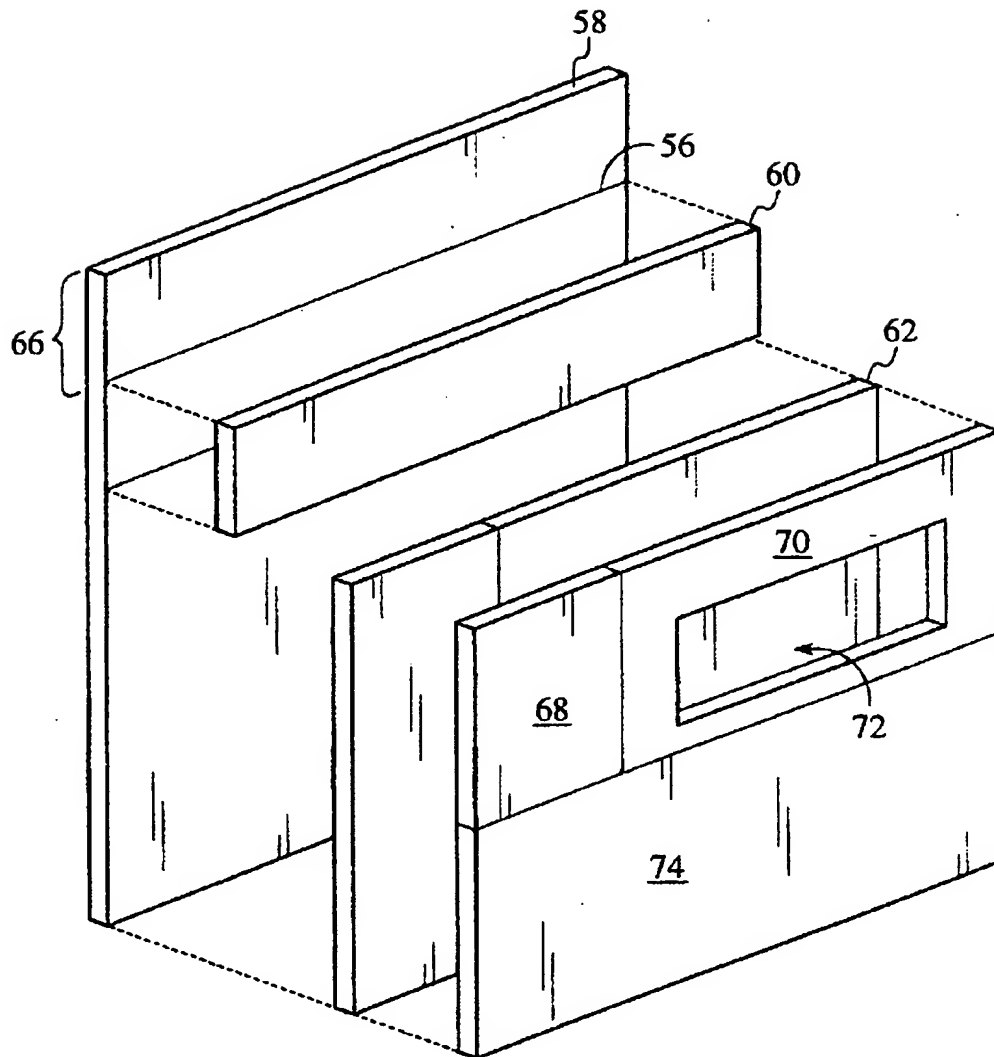


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/02512

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 5/01, 15/00, 17/30, 3/14

US CL : 709/246; 345/326, 356, 349, 342; 707/02

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/246; 345/326, 356, 349, 342; 707/02

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 5,727,159 A (KIKINIS) 10 March 1998, col. 4, lines 15-68 to col. 13, lines 1-68.	1-20
Y, P	US 5,748,188 A (HU et al) 5 May 1998, col. 2, lines 38-68 to col. 26, lines 1-68	1-20
Y, P	US 5,812,134 A (POOSER et al) 22 September 1998, col. 3, lines 1-68 to col. 6, lines 1-44.	1-20
Y, P	US 5,778,356 A (HEINY) 7 July 1998, col. 2, lines 62-68 to col. 3, lines 1-15 and col. 4, lines 51-68 to col. 32, lines 1-25.	1-20

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	*T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A*	document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means		
P document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

21 APRIL 1999

Date of mailing of the international search report

15 JUN 1999

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

HIEU C. LE

Telephone No. (703) 306-3101

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/02512

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A, P	US 5,889,520 A (GLASER) 30 March 1999, col. 2, lines 66-68 to col. 9, lines 1-31.	1-20
A, P	US 5,883,626 A (GLASER et al) 6 March 1999, col. 4, lines 1-68 to col. 7, 1-68.	1-20

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.